13/11/2020

**Experiment No:14**

**QUEUE USING LINKED LIST**

**AIM:**

Write a program to implement Queue using on Linked List.

**DATA STRUCTURES USED:**

Linked List

**ALGORITHM:**

Algorithm INSERT(ITEM)

1. new= GetNodes(Node)
2. new->DATA=ITEM
3. new->LINK=NULL
4. if (new = NULL) then
5. print”memory underflow”
6. Exit
7. Else
8. If (FRONT->LINK=NULL)
9. FRONT->LINK=new
10. REAR->LINK=new
11. else
12. REAR->LINK->LINK= new
13. REAR->LINK=new
14. EndIf
15. EndIf
16. Stop

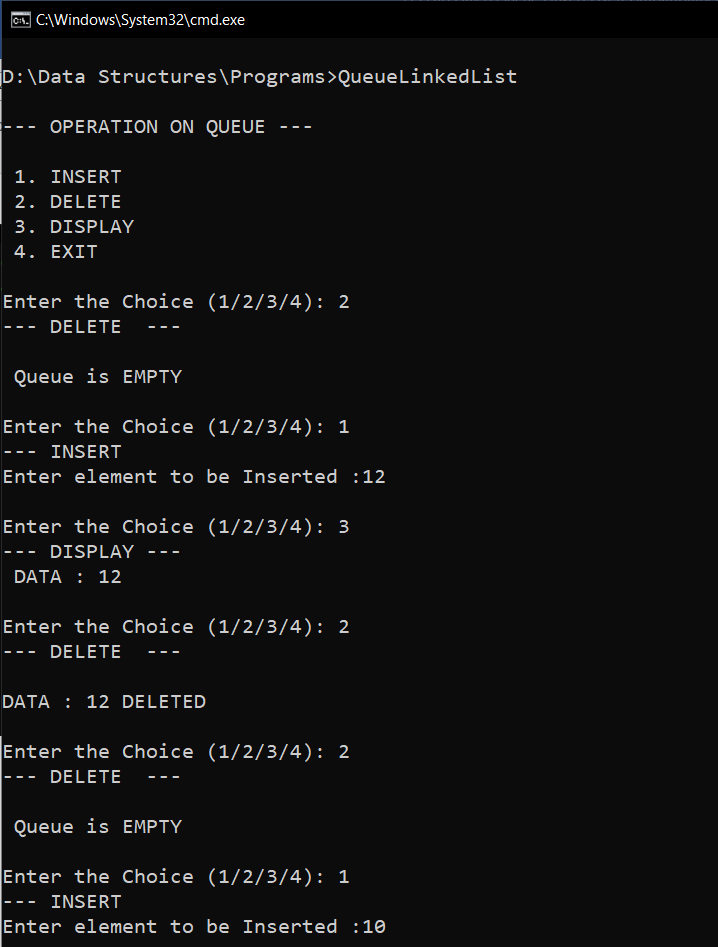
Algorithm DELETE()

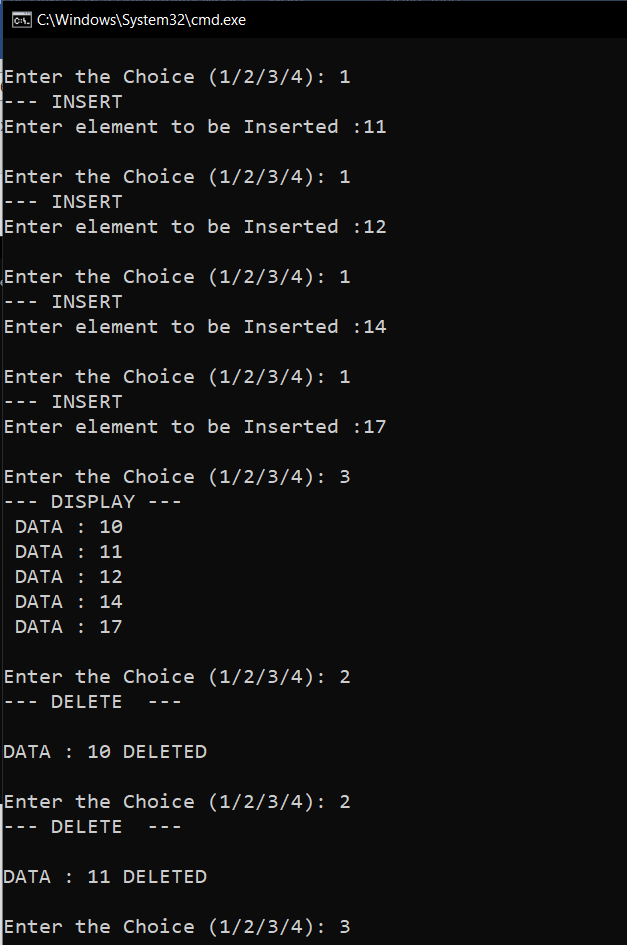
1. ptr=HEADER->LINK
2. if(ptr=NULL)then
3. print “The Queue is empty”
4. Exit
5. Else
6. FRONT->LINK=ptr->LINK
7. ReturnNode(ptr)
8. stop

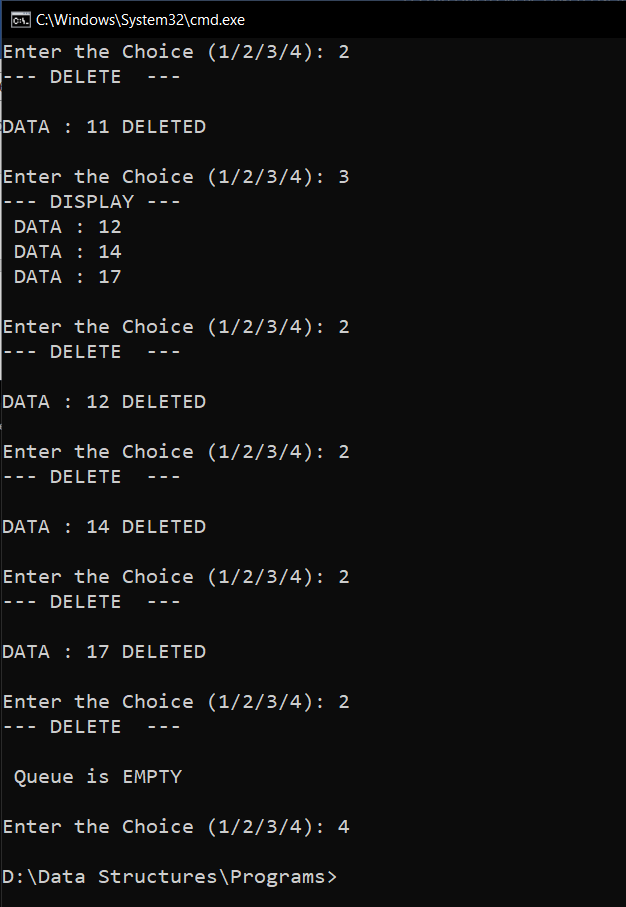
**PROGRAM:**

#include<stdio.h>  
#include<stdlib.h>  
struct node{  
 int data;  
 struct node \*link;  
};  
  
void insert(struct node\* front,struct node\* rear,int x){  
 struct node\* new = (struct node\*)malloc(sizeof(struct node));  
 new->data=x;  
 new->link=NULL;  
 if(new==NULL){  
 printf("\nMEMORY Underflow\n");  
 }else{  
 if(front->link==NULL){  
 front->link=new;  
 rear->link=new;  
 }else{  
 rear->link->link=new;  
 rear->link=new;  
 }  
 }  
}  
void delete(struct node\* front,struct node\* rear){  
 struct node\* ptr=front->link;  
 if(ptr==NULL){  
 printf("\n Queue is EMPTY\n");  
 }else{  
 front->link=ptr->link;  
 printf("\nDATA : %d DELETED\n",ptr->data);  
 free(ptr);  
 }  
}  
void display(struct node\* front,struct node\* rear){  
 struct node\* ptr=front;  
 while(ptr->link!=NULL){  
 ptr=ptr->link;  
 printf(" DATA : %d\n",ptr->data);  
 }  
}  
  
void main(){  
 int n,x,y;  
 char ans='y';  
 struct node\* front = (struct node\*)malloc(sizeof(struct node));  
 struct node\* rear = (struct node\*)malloc(sizeof(struct node));  
 front->link=NULL;  
 rear->link=NULL;  
 printf("\n--- OPERATION ON QUEUE --- \n\n");  
 printf(" 1. INSERT \n");  
 printf(" 2. DELETE \n");  
 printf(" 3. DISPLAY\n");  
 printf(" 4. EXIT\n");  
 while(ans=='y'){  
 printf("\nEnter the Choice (1/2/3/4): ");  
 scanf("%d",&n);  
 switch(n){  
 case 1:printf("--- INSERT \n");  
 printf("Enter element to be Inserted :");  
 scanf("%d", &x);  
 insert(front,rear,x);  
 break;  
 case 2:printf("--- DELETE ---\n");  
 delete(front,rear);  
 break;  
 case 3:printf("--- DISPLAY ---\n");  
 display(front,rear);  
 break;  
 case 4:ans='n';  
 break;  
 default:printf("Enter a Valid Input\n");  
 }  
 }  
}

**OUTPUT:**







**RESULT:**

The Program was successfully compiled and the required output was obtained.

Time complexity of INSERT() operation is O(1).

Time complexity of DELETE() operation is O(1).